Organization
Teaser
As long as there were no machines, programming was no problem at all; when we had a few weak computers, programming became a mild problem, and now we have gigantic computers, programming has become a gigantic problem.

E.W. Dijkstra

The Humble Programmer

The Team
The Team

Dr. Michael Eichberg

Sven Amann

Manuel Weiel

Contact

WWW
http://www.stg.tu-darmstadt.de/teaching/teaching_iverview.en.jsp

Forum
http://www.d120.de/forum/
→kanonische Einführungsveranstaltungen
→Einführung in Software Engineering
Organization
Lecture

- Thursdays 13:30-15:00 in S2 02 | C205 and C110
- The slides are in English
  (Key terms will be translated into German.)
- The slides will generally be available after the lecture
  (I will try hard to make a preliminary version available the day before the lecture.)
- The slides can be found at:
Exercises

• Every week, we will have an exercise, starting today
• Exercises are expected to be solved in teams of 3 students
• The content of the exercise is relevant for the exam
• Sign-up as a team; if you don’t have a team, we will assign you to a team
• All further details will be announced in today’s exercise.
• The content of the exercise is relevant for the exam
Written Exam

• The exam will be on February, 26th 2015 - 14:30 - 16:00
  (The room will be announced in due time. The exam will take 90min.)

• You need to register for the exam in TUCaN
  (There are no further prerequisites; “everyone” can attend the exam.)

• The exam will be an open book exam

• The exam will enable you to choose the tasks that you are
  particular good at
  (I.e., it will not be possible to solve the entire exam in 90 minutes.)
Related Bibliography

2. Writing Effective Use Cases by Alistair Cockburn
3. Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides
4. Applying UML and Patterns An Introduction to Object-Oriented Analysis and Design and the Unified Process by Craig Larman
5. 97 Things Every Programmer Should Know, Collective Wisdom from the Experts

Essential Bibliography

• Design Patterns - Elements of Reusable Object-Oriented Software; Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides; Addison-Wesley, 1995

• Applying UML and Patterns - An Introduction to Object-oriented Analysis and Design; Craig Larman; Prentice Hall
A Recommended / Very Useful Podcast

Recent Episodes

Episode 211: Continuous Delivery on Windows with Rachel Laycock and Max Lincoln
Filed in Episodes by SE-Radio on September 30, 2014 • 0 Comments

Johannes talks with Rachel Laycock and Max Lincoln from ThoughtWorks about continuous delivery on Windows. The outline includes: introduction to continuous delivery, continuous integration: DevOps and ChatOps; decisions to be taken when implementing continuous delivery on windows; build tools on windows; packaging and deploy on windows; infrastructure automation and infrastructure as code with chef, puppet [...]

Continue Reading »

Episode 210: Stefan Tilkov on Architecture and
External Talks / Events

• MSG
  “Requirements Engineering”
  November, 2014

• Capgemini
  Excursion / One-day Workshop
  December, 2014

• ACCSO - Working Title:
  “Beschleunigte Softwareentwicklung”
  January, 2015
The Lecture
Basic programming skills are required.

- Basic knowledge of **object-oriented programming** concepts is necessary. I.e., you should readily understand the following terms:
  - class, interface
  - object
  - inheritance
  - polymorphism
- Working knowledge of the **Java (7)** programming language
- We are going to use Java 8
Basic Goals

• To get a brief overview of “all” areas of software engineering
• To understand agile software development processes
• To be able to perform object-oriented analysis and design
• To be able to read and create basic UML diagrams
• To perform basic Software Quality Assurance
• To get first hands-on experience and to learn to use basic software development tools
The goal is to enable you to systematically carry out small(er) commercial or open-source software projects.
Introduction to Software Engineering

- What is Software Engineering (?)
- Software Project Management
- Software Engineering Tools
- Object-Oriented Analysis and Design
- UML Modeling
Introduction to Software Engineering

Software Project Management

Properties of Software Projects
Requirements Engineering
Software Engineering Processes
Risk Management

Agile Methods
Waterfall Model

Extreme Programming
Unified Process

Introduction to Software Engineering

What is Software Engineering?
Software Project Management
Software Engineering Tools
Object-Oriented Analysis and Design
UML Modeling
Introduction to Software Engineering

Software Engineering Tools

- IDEs
- Revision Control Systems
- Unit Testing

- SVN
- GIT
- ECLEmma
Introduction to Software Engineering

Object-Oriented Analysis and Design

UML Modeling

... requires ...

Class Diagrams
Sequence Diagrams
Communication Diagrams

... requires ...

General Design Goals
Responsibility Driven Design / GRASPrinciples
Design Heuristics
Domain Modeling
Design Patterns

High Cohesion
Low Coupling
Singleton
Template Method
Template Method

What is Software Engineering (7)
Software Project Management
Software Engineering Tools
Object-Oriented Analysis and Design
UML Modeling
Developing Competence

Information + cross-linking

Knowledge + application

Ability + assessment (e.g., the choice of a specific design/process)

Attitude + suitability / adequacy

Competence + responsibility

Professionalism
Content Distribution

"General" Software Engineering Topics:
- Software Quality
- Requirements Engineering
- Software Engineering Processes
- ...

- General Design Principles
- Object-Oriented Design Heuristics
- Object-Oriented Design Patterns